AD-A090 638 CARNEGIE-MELLON UNIV PITTSBURGH PA MANAGEMENT SCIENC--ETC F/G 12/2
ROUND TRIP LOCATION ON A TREE GRAPH.(U)
NOV 79 R V NAGELHOUT N00014-75-C-0621
UNCLASSIFIED MSRR-449 NL

1 OF 1
AD
A090638

END
DATE
FILMED
11 80
DTIC

AD A090638

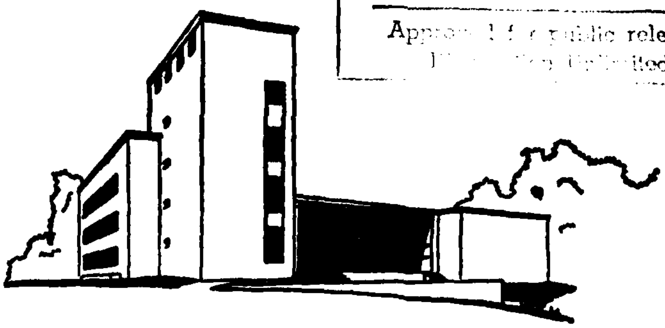
II

Carnegie-Mellon University

PITTSBURGH, PENNSYLVANIA 15213

GRADUATE SCHOOL OF INDUSTRIAL ADMINISTRATION

WILLIAM LARIMER MELLON, FOUNDER



LEVEL II W.P. #37-79-80

9 MANAGEMENT SCIENCES RESEARCH REP[redacted]T NO. 449

6 ROUND TRIP LOCATION ON A TREE GRAPH.

by

Robert V. Nagelhout

Nov[redacted]79

12 12

14 MSRR-449, WP-

15 This report was prepared as part of the activities of the Management Sciences Research Group, Carnegie-Mellon University, under contract N00014-75-C-0621 NR 047-048 with the Office of Naval Research. Reproduction in whole or in part is permitted for any purpose of the U. S. Government.

Management Sciences Research Group
Graduate School of Industrial Administration
Carnegie-Mellon University
Pittsburgh, Pennsylvania 15213

403426

Abstract

In this paper we consider the problem of locating a single facility on a tree graph so as to minimize the maximum "round trip" distance from this facility to a given set of node pairs. The node pairs might represent a neighborhood and a hospital or a neighborhood and a police station to which service must be provided. Each node pair can be assigned a weight which represents the relative amount of service which must be provided at that node pair to the total amount of estimated service. In the case where all of the weights are equal we provide the results necessary to justify a simple $O(m)$ procedure for finding an optimal location, where m is the number of node pairs. When the weights are unequal an $O((n+m)\ell gn + m\ell gm)$ procedure for finding an optimal location is presented where n is the number of nodes in the tree graph. The classical one center problem on a tree graph is a special case of this problem. Examples and references to related research are included.

1. Problem Formulation

Given a connected undirected network, N, with node or vertex set V, and edge or arc set E, let $d(x,y)$ be the length of a shortest path between any two points x and y on N (x and y are not necessarily nodes). Let $\{(p_i,q_i)|p_i \in V,\ q_i \in V,\ 1 \leq i \leq m\}$ be a given set of node pairs. For any point $x \in N$ we define,

$$r_i(x) = d(x,p_i) + d(x,q_i) + k_i \quad 1 \leq i \leq m \tag{1}$$

where $r_i(x)$ is the "round trip" distance from x to node pair i, and k_i is a known constant. In particular if we set $k_i = d(p_i,q_i)$ then $r_i(x)$ represents the length of a shortest path from x to p_i, from p_i to q_i, and then from q_i back to x, thus the name round trip. In the context of this paper we let $k_i = d(p_i,q_i) + k_i'$, so that $r_i(x)$ represents the round trip distance from x to node pair i plus a fixed charge.

Now suppose we associate with each node pair, (p_i,q_i), a positive weight w_i which denotes the importance or relative value of a round trip to node pair i. Next we define,

$$r(x) = \max_{1 \leq i \leq m} \{w_i r_i(x)\} \tag{2}$$

where $r(x)$ is the maximum weighted round trip distance from x to any node pair. Now we consider the problem,

$$P_N: \ \underset{x \in N}{\text{Minimize}}\ r(x)\ . \tag{3}$$

In problem P_N we seek the set of locations, RC, on N such that the maximum weighted round trip distance from any point in RC is

minimized over all possible locations on N. In this paper we address the special case of problem P_N where N is a tree, that is we consider,

$$P_T: \underset{x \in T}{\text{Minimize}} \; r(x) \tag{4}$$

where T is a connected graph with no cycles.

Applications of round trip location can be found in many situations, such as in public sector location problems, where locating a facility with the objective of minimizing the worst case occurrence is a valid criterion. For example consider the location of an ambulance station in an urban center where the edges or arcs of the network correspond to streets and highways, the nodes, p_i, represent neighborhoods where the demand for ambulance service will occur and the nodes, q_i, correspond to local hospitals or service centers. In general a patient picked up by an ambulance in neighborhood p_i would be transported to the nearest hospital and thus node p_i would be paired with a node q_i representing this nearest hospital. However, in some cases the resources necessary to treat a patient, such as equipment, specialists, or drugs, are unavailable at the nearest hospital (this is especially true in the case where local hospitals share resources) and thus it might be necessary to transport the patient to a hospital which is further away. Thus a neighborhood might be paired with several hospitals besides the one nearest to it. In order to distinguish the relative frequency of occurrence or importance of each neighborhood-hospital pair we assign weights to each pair, the larger weights being given to the neighborhood-hospital pairs with the highest priority. Then the objective in P_N is to locate an ambulance station so that the maximum weighted round trip distance

from this ambulance station to a neighborhood-hospital pair is minimized. For other discussions of round trip location see [1] and [2].

In [1], Chan and Francis described an algorithm of complexity $O(m^2)$ for finding an optimal solution to P_T. In section two we provide a brief summary of their approach. Then in section three we provide results which show that for an unweighted tree, a very simple procedure with complexity $O(m)$ can be used to find an optimal solution to P_T. In section four we provide a method for finding an optimal solution to the weighted version of P_T in $O((n+m)\ell gn + m\ell gm)$ (where n is the number of nodes in T) which is based on the concept of a "local" round trip center and the "centroid" of a tree.

Finally we would like to point out that in the special case of P_N where $p_i = q_i$, and $k_i = 0$ for all i, $i \leq i \leq m$, P_N reduces to the well known one center problem which has attracted a considerable amount of attention in the literature. In particular part of the work in [6] and [7] provided strong motivation for the research presented in sections three and four of this paper respectively.

2. Notation and Summary of Previous Results

In this section we introduce some of the notation and definitions which are used throughout the remaining sections of this paper plus a brief summary of some previous results for P_T which can be found in [1].

Notation:

T - a tree; a connected graph with no cycles.

V - the set of nodes of T; $|V| = n$

$\{(p_i, q_i)\}$ - a set of given node pairs $1 \leq i \leq m$

P_i - the (unique) path connecting p_i and q_i

(Note: $P_i = \{z \in T | d(p_i,z) + d(z,q_i) = d(p_i,q_i)\}$

$p_i(x)$ - The point on P_i which is closest to x.

p_{ij} - the point on P_i which is closest to every point on P_j

(Note: If $P_i \cap P_j \neq \emptyset$ then let $p_{ij} = p_{ji}$ be the point on the connected path $P_i \cap P_j$ which is closest to p_i.)

$P(x,y)$ the (unique) path connecting x and y

$V(x)$ - the set of all nodes in V which are adjacent to x

$T_v(x)$ - defined for all $v \in V(x)$, is the connected subtree of T which contains the segment (x,v) plus all the points in T which are connected to x by a path containing v.

RC - the set of all optimal solutions to P_T.

Now we give a brief outline of a procedure for calculating RC, the set of optimal solutions to P_T. First some definitions are necessary. We define

$$g_i = \frac{(d(p_i,q_i) + k_i)}{2} \qquad 1 \leq i \leq m \qquad (5)$$

where g_i represents one half of the fixed charge portion of the round trip distance to node pair i.

Next we note that for all $x \in T$ and for all $y \in P_i$ that $p_i(x) \in P(y,x)$. In particular if $x \in P_i$ then $p_i(x) = x$. Given this property and (5) one can easily verify that $r_i(x)$ can be redefined from (1) as,

$$r_i(x) = 2(d(x,p_i(x)) + g_i) . \tag{6}$$

This alternate definition of $r_i(x)$ will be used in subsequent sections.

Now we define a number, H, which is shown in [1] to equal the optimal objective function value for P_T.

For $1 \leq i \leq j \leq m$ define

$$h_{ij} = \frac{2(w_i w_j D_{ij} + w_i g_j + w_j g_i)}{(w_i + w_j)} \tag{7}$$

and

$$H = h_{st} = \max_{1 \leq i \leq j \leq m} \{h_{ij}\} \tag{8}$$

Note: for $i = j$, $h_{ij} = 2g_i$.

Intuitively H represents the weighted round trip "diameter" of the tree. Since H equals the optimal objective function value for P_T, the set of optimal solutions, RC, can be written as

$$RC = \{x | r_i(x) \leq \frac{H}{w_i} , \quad 1 \leq i \leq m\} \tag{9}$$

or letting $u_i = \dfrac{(H - 2w_i g_i)}{2w_i}$ and considering the definition of $r_i(x)$ in (6) we have

$$RC = \{x | \; d(x,p_i(x)) \leq u_i \quad i \leq i \leq m\} . \tag{10}$$

Remark 1: It is well known (see [3]) that minimizing a convex function, such as $r(x)$, on a tree yields a set of optimal solutions which is convex. By convex on a tree we mean for any

$x_1, x_2 \in T$ let z be the (unique) point which is $\lambda \cdot d(x_1,x_2)$ units from x_1 and $(1-\lambda)\cdot d(x_1,x_2)$ units from x_2, $0 \leq \lambda \leq 1$; then T is convex iff $z \in P(x_1,x_2)$.

Remark 2: Convexity and connectedness are equivalent concepts when defined on a tree.

From Remarks 1 and 2 we see that RC is a convex, or equivalently a connected set on T.

Now the method described in [1] to find RC works as follows: (i) first compute H, and (ii) then compute RC depending upon which of the following two mutually exclusive cases occur,

Case 1: $H = h_{st} > \min(h_{ss},h_{tt}) = \min(2g_s,2g_t)$. In this case $P_s \cap P_t = \emptyset$ and there is a unique optimal solution, x^*, to P_T which satisfies,

(i) $x^* \in P(p_{st},p_{ts})$ and

(ii) $d(p_{st},x^*) = u_s$ and $d(p_{ts},x^*) = u_t$.

That is, x^* is unique and lies on the path between p_{st} and p_{ts} a distance of u_s units from p_{st} or u_t units from p_{ts}.

Case 2: $H = h_{ss} = 2g_s$.

In this case RC is a connected set (see Remarks 1 and 2) which is contained entirely in P_s. More precisely,

$$RC = \{x \in P_s \mid \max_{i \neq s} v_{1i} \leq d(p_s,x) \leq \min_{i \neq s} v_{2i}\} \tag{11}$$

where for $i \neq s$ v_{1i} and v_{2i} are calculating by the following rules. which depend upon whether or not $P_i \cap P_s = \emptyset$.

Case 2(a): $P_i \cap P_s = \emptyset$, then

$$v_{1i} = \max\{d(p_s,p_{si}) - (u_i - D_{si}),0\}$$

$$v_{2i} = \min\{d(p_s,p_{si}) + (u_i - D_{si}),d(p_s,q_s)\} \quad .$$

Case 2(b): $P_i \cap P_s \neq \emptyset$.

Let $P_i \cap P_s = P(a_{si},b_{si})$ where $a_{si} \in P(p_s,b_{si})$ and let

$$v_{1i} = \max\{d(p_s,a_{si}) - u_i, 0\}$$

$$v_{2i} = \min\{d(p_s,b_{si}) + u_i, d(p_s,q_s)\} \quad .$$

The method is illustrated for Cases 1 and 2 in the following examples which are taken from [1].

Example 1: See Figure 1. $g_1 = 6$, $g_2 = 7$, $g_3 = 6$, $D_{12} = 0$, $D_{13} = 2$, $D_{23} = 2$, $h_{12} = g_1 + g_2 = 13$, $h_{13} = 2(2 + 6 + 6)/2 = 14$, $h_{23} = 2(2 + 7 + 6)/2 = 15$. Therefore $H = h_{23} = 15$. This is Case 1 $u_3 = (15 - 12)/2 = 1.5$, $u_2 = (15 - 14)/2 = .5$ and x^* is the unique point, as shown in Figure 1, .5 units from a and 1.5 units from b.

Example 2: See Figure 2. $g_1 = 8$, $g_2 = 7$, $g_3 = 5$, $D_{12} = 0$, $D_{13} = 2$, $D_{23} = 2$, $h_{12} = g_1 + g_2 = 15$, $h_{13} = 2(2 + 8 + 5)/2 = 14$, $h_{23} = 2(2 + 7 + 5)/2 = 14$ therefore $H = h_{11} = 2g_1 = 16$. This is Case 2; $u_2 = (16 - 14)/2 = 1$, $u_3 = (16 - 10)/2 = 3$. For $i = 2$, $P_1 \cap P_2 = P(p_2,a)$ which is Case 2(b), $v_{12} = \max\{3 - 1, 0\} = 2$ and $v_{22} = \min\{6 + 1, 8\} = 7$. For $i = 3$, $P_1 \cap P_3 = \emptyset$. This is case 2(a) thus $p_{13} = a$ and $v_{13} = \max\{6 - (3 - 2), 9\} = 5$, $v_{23} = \min\{6 + (3 - 2), 8\} = 7$.

$$RC = \{x \in P_1 : \max\ (2,5) \le d(p_1,x) \le \min\ (7,7)\}$$

$= \{x \in P_1 : 5 \le d(p,x) \le 7\}$. RC is illustrated in in Figure 2.

Once the distance matrix has been determined the computational effort involved in finding RC is $O(m^2)$ operations to calculate H plus whatever is necessary to calculate the v_{ij}, a_{si}, and b_{si}. In the next section we provide results which justify an $O(m)$ method for locating an optimal solution to the unweighted version of P_T.

3. Finding an Optimal Facility in an Unweighted Tree

Suppose that $w(i) = c$ for all i $1 \le i \le m$ or without loss of generality that $c = 1$. Given $x \in T$ and $v \in V(x)$ recall the definition of $T_v(x)$ from Section 2 as the connected subtree of T which contains the segment (x,v) plus all $y \in T$ such that $v \in P(y,x)$. The validity of the following remark is based on the fact that T is a tree.

Remark 3: $p_i \in T_v(x)$ and $q_i \in T_v(x)$ if and only if $P_i \subseteq T_v(x)$.

Definition 1: $L(x) = \{\ell_1(x),\ldots,\ell_{k(x)}(x)\}$ is the set of indices of the node pairs $(p_{\ell_i(x)}, q_{\ell_i(x)})$ where the maximum in calculating $r(x)$ from (2) occurs. $L(x)$ represents the set of bottleneck node pairs with respect to x and $P_{\ell_i(x)}$ for each i, $1 \le i \le k(x)$, is a bottleneck path with respect to x.

Although we assume in this section that $w(i) = 1$ for all i, $1 \le i \le m$, the following lemmas 1 and 2 hold true for any $w(i) > 0$. We state and prove the lemmas using any positive weights since the more general results will be used later in Section 4 in the weighted version of P_T.

<u>Lemma 1</u>: *Given* $x \in T$ *suppose that for some* $\ell_i(x) \in L(x)$, $1 \leq i \leq k(x)$, and for some $v \in V(x)$ that $P_{\ell_i(x)} \subseteq T_v(x)$. Then $RC \subseteq T_v(x)$.

Proof: Suppose there exists $x^* \in RC - T_v(x)$. Let r^* denote the optimal round trip distance (i.e., r^* is the optimal objective function value of P_T when $w(i) > 0$ for all i). For notational convenience let $s = \ell_i(x)$. Since $P_s \subseteq T_v(x)$ we must have $P_s(x^*) \in T_v(x)$, and since by assumption $x^* \notin T_v(x)$ we have

$$\text{(i)} \quad x \in P(x^*, p_s(x^*)) \quad \text{and} \quad \text{(ii)} \quad p_s(x^*) = p_s(x).$$

Consequently,

$$\begin{aligned}
r(x^*) &= r^* \geq w(s)(d(x^*,p_s) + d(x^*,q_s) + k_s) && \text{by definition of } r^* \\
&= 2w(s)\cdot(d(x^*,p_s(x^*)) + g_s) && \text{from (6)} \\
&= 2w(s)\cdot(d(x^*,x) + d(x,p_s(x^*) + g_s) && \text{from (i)} \\
&= 2w(s)\cdot(d(x^*,x) + d(x,p_s(x) + g_s) && \text{from (ii)} \\
&= 2w(s)\cdot d(x^*,x) + r(x) && \text{from (6) and } s \in L(x) \\
&> r(x) && \text{since } x^* \neq x
\end{aligned}$$

which contradicts the assumption that $x^* \in RC$. $\square$

<u>Corollary</u>: Given $x \in T$ suppose that for some $v_1, v_2 \in V(x)$, $v_1 \neq v_2$, and some i,j, $1 \leq i \leq j \leq k(x)$, that $P_{\ell_i(x)} \subseteq T_{v_1}(x)$ and $P_{\ell_j(x)} \subseteq T_{v_2}(x)$. Then x is the unique optimal solution to P_T.

Lemma 2: Given $x \in T$ and any $\ell_i(x) \in L(x)$ suppose that $x \in P_{\ell_i(x)}$, then

(i) $x \in RC$ and

(ii) $RC \subseteq P_{\ell_i(x)}$.

Proof: Let $s = \ell_i(x)$. Consider any point $y \neq x$. We have

$$
\begin{aligned}
r(y) &\geq w(s) r_s(y) && \text{by definition of } r(y) \\
&= 2w(s)\cdot(d(y,p_s(y)) + g_s) && \text{from (6)} \quad (12) \\
&\geq 2w(s)\cdot(d(x,p_s(x)) + g_s) && \text{since } p_s(x) = x \quad (13) \\
&= r(x) && \text{since } s \in L(x).
\end{aligned}
$$

This proves (i). From lines (12) and (13) it is evident that if $d(y,p_s(y)) > d(x,p_s(x)) = 0$, then $r(y) > r(x)$. Thus $r(y) = r(x)$ only if $d(y,p_s(y)) = 0$ or equivalently only if $y \in P_s$, which proves (ii). □

Lemma 2 says that if $x \in P_{\ell_i(x)}$ for some $\ell_i(x) \in L(x)$, then x is an optimal location and the set of all optimal locations for P_T is contained in the bottleneck path $P_{\ell_i(x)}$.

Lemma 3: $r(x) \leq r(y) + 2d(x,y) \quad \forall\, x,y \in T$.

Proof: We have from the definition of $r(y)$

$$r_i(y) = d(y,p_i) + d(y,q_i) + k_i \leq r(y) \qquad \forall\, 1 \leq i \leq m \tag{14}$$

Also, making use of the triangle inequality which is valid for trees we have for all i,

$$d(x,p_i) + d(x,q_i) + k_i \leq d(x,y) + d(y,p_i) + d(x,y) + d(y,q_i) + k_i \tag{15}$$

Combining (14) and (15) we get for all i,

$$d(x,p_i) + d(x,q_i) + k_i \leq 2d(y,x) + r(y) \tag{16}$$

and finally since $r(x) = \max_{1\leq i\leq m} \{d(x,p_i) + d(x,q_i) + k_i\}$ we get the desired result. $\square$

Now we would like to define what we shall call an "extreme point" and an "interior point" of RC.

<u>Definition 2</u>: A point $x \in RC$ is an <u>extreme point</u> of RC if and only if there exists a number $\delta > 0$, δ arbitrarily small, and a point $y \notin RC$ such that (i) $d(y,x) = \delta$ and (ii) $(P(y,x) - \{x\}) \cap RC = \emptyset$.

<u>Remark 4</u>: From lemmas 1 and 2 and their corallaries it is clear that RC is either a connected set which is contained in a bottleneck path or a single point. In any event RC can be written as,

$$RC = P(a^*,b^*) = \{x | d(a^*,x) + d(x,b^*) = d(a^*,b^*)\} \tag{17}$$

where a^* and b^* are the endpoints of RC.

From Definition 2 and Remark 4 it is clear that if $x \in E$ then $x = a^*$, $x = b^*$, or $x \in V$, which leads us to the conclusion $E \subseteq (RC \cap V) \cup \{a^*,b^*\}$.

<u>Definition 3</u>: A point $x \in I = RC - E$ is called an <u>interior point</u> of RC.

Next we would like to associate with each point $x \in T$ a point in RC, which we call $c(x)$. We define $c(x)$ as follows:

Definition 4: Let $c(x)$ be the unique point in RC which is closest to x, that is $c(x)$ satisfies

$$d(x,c(x)) = \min_{y \in RC} \{d(x,y)\} ,$$

Note that if $x \in RC$ then $c(x) = x$.

Definition 5: For all $x \in T-I$ let v_x be defined as any node in $V(c(x))$ which satisfies the following two properties:

(i) $x \in T_{v_x}(c(x))$

(ii) $RC \subseteq (T - T_{v_x}(c(x))) \cup \{c(x)\}$.

Later, in Lemma 5, we show that v_x is well defined.

For an example of the previous definitions see Figures 1 and 2. In Figure 1 $E = RC = \{x^*\}$ and $I = \emptyset$; $c(x) = x^*$ $\forall x \in T$; for each $x \in T_a(x^*)$, $v_x = a$; for each $x \in T_b(x^*)$, $v_x = b$.

In Figure 2 RC is shown: $E = \{a^*, a, b^*\}$; for each $x \in T_b(a)$, $c(x) = a$ and $v_x = b$; for each $x \in T_{p_2}(a^*)$, $c(x) = a^*$ and $v_x = p_2$; for each $x \in T_{q_2}(a)$, $c(x) = a$ and $v_x = q_2$; and for each $x \in T_{q_1}(b^*)$, $c(x) = b^*$ and $v_x = q_1$.

In the next lemma we provide an alternate characterization of the extreme points of RC which in turn leads to a proof that for all $x \in T-I$, v_x is well defined.

Lemma 4: $E = \{x \in RC | x = c(y) \text{ for some } y \in T-I\}$.

Proof: ($\supseteq$) Let $x = c(y)$ for some $y \in T-I$. Now if $y \in T$ then by definition $x = c(y) = y \in E$. Suppose that $y \notin E$. Then x is the closest point in RC to y. Thus we have (i) $d(x,y) > 0$ and (ii) $(P(x,y) - \{x\}) \cap RC = \emptyset$ which shows from Definition 2 that $x \in E$.

($\subseteq$) Let $x \in E$. From Definition 2 we see that there exists a point $y \notin RC$ such that $(P(y,x) - \{x\}) \cap RC = \emptyset$. Since RC is connected and T is a tree this implies that x is the closest point in RC to y. $\square$

<u>Lemma 5</u>: For all $x \in T\text{-}I$, v_x is well defined (that is v_x exists).

Proof: There are two cases to consider,

<u>Case 1:</u> $x \in E$.

Then $c(x) = x \in E$ and from Definition 2 there exists $y \notin E$ which is arbitrarily close to x. Since y is arbitrarily close there exists $v \in V(y)$ such that $v \in V(x)$ and $y \in P(v,x)$. Clearly $x \in T_v(x)$ and this proves (i) of Definition 5. Next since $x \in E$ we have $(P(y,x) - \{x\}) \cap RC = \emptyset$ and since RC is connected we must have $RC \subseteq (T - T_v(x)) \cup \{x\}$, which proves (ii).

<u>Case 2</u>: $x \notin E$, and since $x \in T\text{-}I \Rightarrow x \notin RC$

Since $x \neq c(x)$ and $P(x,c(x)) \cap RC = \{c(x)\}$ there exists a $y \in P(x,c(x))$ such that y is arbitrarily close to $c(x)$. We use the same argument as in Case 1 to prove (i) and then recalling from Lemma 4 that $c(x) \in E$ we again use the same argument as in Case 1 to prove (ii). $\square$

<u>Remark 5</u>: Although Lemma 5 shows that v_x exists for all $x \in T\text{-}I$, it is not necessarily true that v_x is unique. For example consider $x = a$ in Figure 2. Here $v_a = q_2$ or $v_a = b$ will satisfy the properties of Definition 5. It can be shown however that if $x \notin RC$, then v_x is unique.

We are now ready to prove the following important result.

<u>Lemma 6</u>: For all $x \in E$ there exists an i, $1 \leq i \leq k(x)$, and $\ell_i(x) \in L(x)$ such that $P_{\ell_i(x)} \subseteq (T - T_{v_x}(x)) \cup \{x\}$ for all $v_x \in V(x)$.

Proof: Let $s = \ell_i(x)$. Since $x \in E$ there exists $y \notin RC$ which is arbitrarily close to x such that $x = c(y)$. Now suppose to the contrary that for all i, $1 \leq i \leq k(x)$, and all $v_x \in V(x)$ we have $P_s \nsubseteq (T - T_{v_x}(x)) \cup \{x\}$. Then we intend to show that there exists a point, x_ε, with the property that (i) $x_\varepsilon \in RC$ and (ii) $d(y, x_\varepsilon) < d(y, x)$. This of course contradicts the fact that $x = c(y)$.

First without loss of generality we assume that,

$$p_s \in T_{v_x}(x) - \{x\} \text{ for all } i \quad 1 \leq i \leq k(x) \tag{18}$$

$$\text{and for all } v_x \in V(x).$$

Otherwise $p_s \in (T - T_{v_x}(x)) \cup \{x\}$ and $q_s \in (T - T_{v_x}(x)) \cup \{x\}$ by Remark 1 we would have $P_s \subseteq (T - T_{v_x}(x)) \cup \{x\}$ which contradicts the supposition of the proof.

Now given any $v_x \in V(x)$ we partition $L(x)$ into two sets depending upon the following two cases:

<u>Case 1</u>: $q_s \in T_{v_x}(x) - \{x\}$.

Given (18) and Remark 1 we have

$P_s \subseteq T_{v_x}(x) - \{x\}$ and thus

$$\delta_s = d(x, p_s(x)) > 0. \tag{19}$$

Let $L_1(x) = \{s \in L(x) | \delta_s > 0\}$.

<u>Case 2</u>: $q_s \in (T - T_{v_x}(x)) \cup \{x\}$.

Given (18) this implies that $x \in P_s$ and $p_s(x) = x$.

In this case we let

$$L_2(x) = \{s \in L(x) | \delta_s = 0\}.$$

Clearly $L(x) = L_1(x) \cup L_2(x)$ and $L_1(x) \cap L_2(x) = \emptyset$.

Now we define several numbers which will be used in verifying the fact that x_ε has the desired properties (i) and (ii) stated previously.

Let $\varepsilon_1 = \min\limits_{i \notin L(x)} \{(r^* - r_i(x))/2\} > 0$

ε_1 equals one half of the minimum difference between r^*, the optimal round trip distance, and the round trip distance from x to all non-bottleneck node pairs.

Let $\varepsilon_2 = \min\limits_{i \in L_1(x)} \{\delta_i\} > 0$

ε_2 equals the minimum distance between x and any bottleneck path which is completely contained in $T_{v_x}(x) - \{x\}$.

Let $\varepsilon_3 = \min\limits_{i \in L_2(x)} \{d(x, p_i)\} > 0$

ε_3 equals the minimum distance from x to p_i over all $i \in L(x)$ such that $x \in P_i$.

Now choose $y \notin RC$, so that $y \in T_{v_x}(x)$ and $d(y,x) = \varepsilon_4 < \min\{\varepsilon_1, \varepsilon_2, \varepsilon_3, d(v_x, x)\}$. Note that this can be done since $x \in E$.

Next let $\epsilon = \frac{\epsilon_4}{2}$ and let x_ϵ be the unique point in $P(y,x)$ which is ϵ units from x.

Claim: $x_\epsilon \in RC$.

We will prove the claim by showing that $r_i(x_\epsilon) \leq r^*$ for each i, $1 \leq i \leq m$.

Case 1: $i \notin L(x)$.

(a) $p_i(x_\epsilon) \in T_{v_x}(x) - \{x\}$.

Since $p_i(x_\epsilon) \in V$ and $x_\epsilon \in P(v_x,x)$ it must be true that $p_i(x_\epsilon) = p_i(x)$ and $d(p_i(x_\epsilon),x_\epsilon) < d(p_i(x_\epsilon),x)$, which implies from (6) that $r_i(x_\epsilon) \leq r_i(x) \leq r^*$.

(b) $p_i(x_\epsilon) \in (T-T_{v_x}(x)) \cup \{c(x)\}$.

Then $x \in P(x_\epsilon,p_i(x_\epsilon))$ which implies that $p_i(x_\epsilon) = p_i(x)$.

We have

$$\begin{aligned} r_i(x_\epsilon) &= 2(d(x_\epsilon,p_i(x_\epsilon) + g_i) && \text{from (6)} \\ &= 2(d(x_\epsilon,x) + d(x,p_i(x_\epsilon)) + g_i) && \text{since } x \in P(x_\epsilon,p_i(x_\epsilon)) \\ &= 2(d(x_\epsilon,x) + d(x,p_i(x) + q_i) && \text{since } p_i(x_\epsilon) = p_i(x) \\ &\leq 2(\frac{(r^*-r_i(x))}{2} + \frac{r_i(x)}{2}) = r^*. \end{aligned}$$

Case 2: $i \in L(x)$.

(a) $i \in L_1(x)$

Since $x_\epsilon \in P(x,p_i,(x))$ we have $p_i(x) = p_i(x_\epsilon)$ and $d(x_\epsilon,p_i(x_\epsilon) < d(x,p_i(x))$ and thus from (6) $r_i(x_\epsilon) < r_i(x) = r^*$.

(b) $i \in L_2(x)$

Since $\epsilon < \epsilon_3$ we have $x_\epsilon \in P_i$ and thus $r_i(x_\epsilon) = r_i(x) = r^*$.

Finally since $\epsilon < \epsilon_4$ we have $x_\epsilon \in P(y,x)$. The existence of x_ϵ contradicts the fact that $x \in E$. $\square$

In the next lemma we establish a relationship between $r(x)$, $c(x)$, and r^* for all $x \in T$.

Lemma 7: For all $x \in T$ $r(x) = r^* + 2d(x,c(x))$.

Proof: If $x \in RC$, then $c(x) = x$ (See Definition 4) and thus the proposition is true. If $x \notin RC$ then from Lemma 4 $c(x) \in E$. From Lemma 6 there exists an i, $1 \leq i \leq k(c(x))$, $\ell_i(c(x))$, and $v_{c(x)} \in V(c(x))$ such that $P_{\ell_i(c(x))} \subseteq (T - T_{v_{c(x)}}(c(x))) \cup \{c(x)\}$ and $x \in T_{v_c(x)}(c(x))$. Let $s = \ell_i(c(x))$. Now since $x \in T_{v_c(x)}(c(x))$ and $p_s(c(x)) \in (T - T_{v_{c(x)}}(c(x))) \cup \{c(x)\}$ we must have

(i) $c(x) \in P(x,p_s(c(x)))$ and (ii) $p_s(c(x)) = p_s(x)$.

From this we get,

$$
\begin{aligned}
r(x) &= r_s(x) && \text{by definition of } r(x)\\
&= 2(d(x,p_s(x))) + g_s && \text{from (6)}\\
&= 2(d(x,c(x)) + d(c(x),p_s(c(x))) + g_s && \text{from (i) and (ii)}\\
&= 2d(x,c(x)) + r(c(x)) && \text{from (6) and } s \in L(c(x))\\
&= 2d(x,c(x)) + r^* && \text{since } c(x) \in RC.
\end{aligned}
$$

Combining this with Lemma 3, setting $y = c(x)$, we get $r(x) \leq r(c(x)) + 2d(x,c(x)) = r^* + 2d(x,c(x))$ which yields the desired result. □

Lemma 7 tells us that for any location $x \in T$, the round trip value, $r(x)$, of this location is linearly proportional to the distance of this location from RC. Thus all of the points in RC have the same round trip value and the further we move from RC, the larger the round trip value becomes. This behavior is indicative of the fact that $r(\cdot)$ is a convex function on T. RC represents the only set of local and global optimal solutions. This observation helps in the proof of the next lemma.

Lemma 8: For any point $x \in T$ and every i, $1 \leq i \leq k(x)$, $c(x) \in P(x, p_{\ell_i(x)}(x))$.

Proof: Let $s = \ell_i(c(x))$. If $x \in RC$, then $c(x) = x$ and thus the proposition is trivially true. Now suppose that $x \notin RC$. From Lemma 6 there exists an s and $v_{c(x)}$ such that

(i) $P_s \equiv (T - T_{v_{c(x)}}(c(x))) \cup \{c(x)\}$ and $x \in T_{v_{c(x)}}(c(x))$.

(See Figure 3). We have at this point (ii) $c(x) \in P(x, p_s(c(x)))$.

Now if we can show that $s \in L(x)$, then this will be one path which satisfies the hypothesis of the lemma. To show this we have,

$$\begin{aligned} r_s(x) &= 2d(x,c(x)) + r_s(c(x)) && \text{from (i) and (ii)} \\ &= 2d(x,c(x)) + r^* && \text{since } c(x) \in RC \\ &= r(x) && \text{from Lemma 7.} \end{aligned}$$

Thus $x \in L(x)$ and $P(x,p_s(c(x))) = P^*$ is a path

which satisfies the hypothesis of the lemma.

Now we use P^* to show that $c(x) \in P(x,p_i(x))$ for all $i \in L(x)$. Suppose to the contrary that there exists a $j \in L(x)$ such that $c(x) \notin P(x,p_j(x))$. Then there exists a node $q \in P(x,c(x))$ where P^* and $P(x,p_j(x))$ separate. (See Figure 3). Then we have

$$\begin{aligned} r(x) &= 2(d(x,q) + d(q,p_j(x)) + g_j) && \text{since } j \in L(x) \quad (20) \\ &= 2(d(x,q) + d(q,c(x)) + r^* && \text{from lemma 7} \quad (21) \end{aligned}$$

Rearranging (20) and (21) we get

$$\begin{aligned} r(c(x)) = r^* &= 2(d(q,p_j(x)) + g_j - d(q,c(x))) \\ &< 2(d(q,p_j(x)) + g_j + d(q,c(x))) && \text{since } d(q,c(x)) > 0 \\ &= r_j(c(x)). \end{aligned}$$

But this contradicts the definition of $r(\cdot)$. □

Lemma 8 says that $c(x)$ is an element of the path from x to the path between every bottleneck node pair of x. This property is crucial in the proof of the following theorem which provides a simple method for solving the unweighted version of P_T.

<u>Theorem 1</u>:

Step 1: Choose any $x_1 \in T$

Step 2: Calculate $r_i(x_1)$ for each i $1 \le i \le m$

Choose any $\ell_i(x_1) \in L(x_1)$ and find $p_{\ell_i(x_1)} = x_2$

Step 3: Calculate $r_i(x_2)$ for each i $1 \le i \le m$

Choose any $\ell_i(x_2) \in L(x_2)$ and find $p_{\ell_i(x_2)} = x_3$.

Let x^* be the unique point on $P(x_2,x_3)$ such that

$$d(x_3,x^*) = \frac{g_{\ell_i(x_1)} + d(x_2,x_3) - g_{\ell_i(x_2)}}{2}$$

Then x^* calculated as in Steps 1 - 3 is optimal.

Proof: We plan to show that $x^* \in P(c(x_2), c(x_3))$ and since RC is a connected set this implies that $x^* \in RC$.

First note by construction that,

$$r_{\ell_i(x_2)}(x^*) = 2(d(x^*,x_3) + g_{\ell_i}(x_2) \qquad \text{since } x^* \in P(x_2,x_3)$$

$$= 2 \frac{g_{\ell_i}(x_1) + d(x_2,x_3) - g_{\ell_i}(x_1)}{2} + g_{\ell_i}(x_2) \qquad \text{from Step 3}$$

$$= 2 \frac{(g_{\ell_i}(x_2) + d(x_2,x_3) - g_{\ell_i}(x_1) + g_{\ell_i}(x_1))}{2}$$

$$= 2(d(x^*,x^2) + g_{\ell_i}(x_1)) = r_{\ell_i(x_1)}(x^*) \; . \qquad (22)$$

Now from Lemma 8 we have $c(x_2) \in P(x_2,x_3)$ and since RC is connected and T is a tree, we have $c(x_3) \in P(x_2,x_3)$. By construction $x^* \in P(x_2,x_3)$. We claim that $x^* \in P(c(x_2),c(x_3))$. Suppose to the contrary that $x^* \notin P(c(x_2),c(x_3))$. Let us first assume that $x^* \in P(c(x_2),x_2) - \{c(x_2)\}$. See Figure 4.

From (22) we have

$$r_{\ell_i(x_1)}(x^*) = r_{\ell_i(x_2)}(x^*) = 2d(x^*,c(x_2)) + r^* \qquad \text{from Lemma 7 and since } c(x^*) = c(x_2) \qquad (23)$$

However since $x^* \in P(c(x_2),x_2) - \{c(x_2)\}$,

$$r_{\ell_i(x_1)}(c(x_2)) = 2d(c(x_2),x^*) + r_{\ell_i(x_1)}(x^*)$$

$$= 4d(c(x_2),x_2^*) + r^* \qquad \text{from (23)}$$

$$> r^* \qquad \text{since } d(c(x_2),x^*) > 0.$$

But this contradicts the fact that $c(x_2) \in RC$. The proof for the case where $x^* \in P(c(x_3),x_3) - \{c(x_3)\}$ is identical. □

Example 3: We perform the procedure of the theorem on the problems in Figures 1 and 2. First in Figure 1:

Step 1 Choose $x_1 = p_1$

Step 2 $r_1(p_1) = 12$; $r_2(p_1) = 16$; $r_3(p_1) = 24$;

$L(p_1) = \{3\}$; $x_2 = p_3(p_1) = b$.

Step 3 $r_1(b) = 16$; $r_2(b) = 18$; $r_3(b) = 12$

$L(b) = \{2\}$; $x_3 = p_2(b) = a$

$x^* \in P(a,b)$, $\frac{6 + 2 - 7}{2} = .5$ units from a.

In Figure 2:

Step 1 Choose $x_1 = p_1$

Step 2 $r_1(p_1) = 16$; $r_2(p_1) = 20$; $r_3(p_1) = 26$

$L(p_1) = \{3\}$; $x_2 = p_3(p_1) = b$

Step 3 $r_1(b) = 20$; $r_2(b) = 24$; $r_3(b) = 10$;

$L(b) = \{2\}$; $x_3 = p_2(b) = a$.

$x^* \in P(a,b)$, $\frac{5 + 2 - 7}{2} = 0$ units from a.

The computational effort involved in calculating x^* is $O(m)$ which is essentially the amount of work which is required to calculate $r_i(x_1)$ in Step 2 and $r_i(x_2)$ in Step 3. In the next section we discuss a method for solving the weighted version of P_T.

4. Round Trip Location on a Weighted Tree

We assume that $w(i) > 0$ for all i, $1 \leq i \leq m$. First we present a method for locating a single edge of T which is guaranteed to contain an optimal solution to P_T. Then we provide a method for finding a "local" solution to P_T, that is a solution to P_T where a

location is constrained to lie on a single edge of the tree. Combining these two approaches yields an algorithm for P_T.

4.1 Locating an Optimal Edge

Using Lemmas 1 and 2, which were stated and proven for the weighted version of P_T, we devise an algorithm which results in exactly one of the following states,

(i) We find a node $v \in V$ which satisfies the properties of Lemma 2 and therefore is optimal.

or (ii) We find a single edge $e = (v_s, v_t)$ of T which is guaranteed to contain an optimal solution to P_T.

In case (ii) we show in Section 4.2 how to find an optimal solution on the edge (v_s, v_t).

First we state the algorithm for (i) and (ii), and then we justify its validity by reference to Lemmas 1 and 2. We assume that T contains at least one edge.

Algorithm 4.1

1. Set $k = 0$, $T_0 = T$.
2. If T_k consists of only the single edge, (v_s, v_t), then go to 5. Otherwise choose $v_k \in V$ such that $|V(v_k)| > 1$. (In Algorithm 4.2 to be stated later we specify more precisely the method for choosing v_k.)
3. Calculate $L(v_k)$. Pick $\ell_i(v_k) \in L(v_k)$, $1 \leq i \leq k(v_k)$. If $v_k \in P_{\ell_i(v_k)}$ then go to 4. Otherwise find $\bar{v}_k \in V(v_k)$ such that $P_{\ell_i(v_k)} \subseteq T_{\bar{v}_k}(v_k)$. Let $T_{k+1} = T_k \cap T_{\bar{v}_k}(v_k)$. Replace k by $k + 1$. Go to 2.
4. Stop. $v_k \in RC$.
5. Stop. $RC \cap (V_s, V_t) \neq \emptyset$.

At iteration k of Algorithm 4.1 we choose a node $v_k \in V$ which is not a "leaf" of T. This is so that in Step 3, $T_{\bar{v}_k}(v_k) \neq T$, i.e. so that $T_{\bar{v}_k}(v_k)$ is a proper subtree of T. This is essential for the convergence of Algorithm 4.1 to a single edge. In Step 3 if $v_k \in P_{\ell_i(v_k)}$ then by Lemma 2 $v_k \in RC$, and we go to Step 4. Otherwise from Lemma 1, $RC \subseteq T_{\bar{v}_k}(v_k)$ which justifies the reduction of T_k to T_{k+1}. Eventually we reach either Step 4 or Step 5.

The computational complexity of Algorithm 4.1 is essentially of $O(m)$ each time Step 3 is performed, which is the effort required to calculate $L(v_k)$ and $T_{\bar{v}_k}(v_k)$. Next we present an algorithm of complexity $O(n)$ for choosing v_k which in turn guarantees that Step 3 will be executed at most $O(\ell gn)$ times, thus making the total computational effort of Algorithm 4.1 $O((n+m)\ell gn)$. Since in most practical cases $m > n$, i.e. the number of node pairs tends to exceed the number of nodes, the complexity would then be $O(m\, \ell g\, m)$.

<u>Definition 6</u>: For any $v \in V$ and $\bar{v}_k \in V(v)$, $1 \leq k \leq |V(v)|$ let $N(v) = \max_{1 \leq k \leq |V(v)|} \{|V \cap T_{\bar{v}_k}(v)|\}$. $N(v)$ is the maximum number of nodes in any of the subtrees $T_{\bar{v}_k}(v)$. A <u>centroid</u>, v^*, of T is a node with the property that,

$$N(v^*) = \min_{v \in V}\{N(v)\} \tag{24}$$

Clearly $N(v^*) \leq \left\lceil \frac{n}{2} \right\rceil + 1$, thus if at each iteration of Step 2 we choose a centroid of T_k, then we have $|T_{k+1} \cap V| \leq \left\lceil \frac{|T_k \cap V|}{2} \right\rceil + 1$. This implies that T_k will be reduced to an edge of T in at most $O(\ell gn)$ operations, so that Step 3 will be executed at most $O(\ell gn)$ times.

In his work on the one median problem Goldman [5] devised an $O(n)$ algorithm for finding a centroid of a tree. Kariv and Hakimi [6] also use this algorithm for the one center problem. We state the algorithm here:

Algorithm 4.2:

(For finding a centroid of T. To be used in Step 2 of Algorithm 4.1 for choosing v_k.)

1. Let $T^1 = T$. For each $v \in V$ let $N(v) = 1$.
2. If T^1 equals a single node, v^*, then stop, v^* is a centroid of T.
3. Choose $v \in T^1 \cap V$ such that $|T^1 \cap V(v)| = 1$. If $N(v) > \frac{n}{2}$ then stop, v is a centroid of T. Otherwise find $u \in V(v) \cap T^1$. Replace $N(u)$ by $N(u) + N(v)$. Remove v and edge (u,v) from T^1, Go to 2.

The proof of the validity of Algorithm 4.2 is omitted here.

Algorithm 4.1 combined with Algorithm 4.2 guarantees a reduction in the search for an optimal solution to an edge of T in $O((n+m)\ell g\ n)$ operations.

4.2 Finding a Local Round Trip Center.

After having located, in Step 5 of Algorithm 4.1, a single edge which is guaranteed to contain an optimal solution, we show in this section how to find an optimal solution on this edge.

We assume that $w(1) \geq \ldots \geq w(m)$ which requires $O(m\ \ell g\ n)$ operations. Let $x_t(e)$ denote the point on edge $e = (v_s, v_t)$ which is t units from v_s. For notational convenience let $d(v_s, v_t) = d$. We seek a point $x_{t^*}(e)$ such that,

$$r(x_{t^*}(e)) = \min_{0 \leq t \leq d} \{r(x_t(e))\} \tag{25}$$

Now first we note that $r_i(x_t(e))$ can be redefined from (6) as

$$\begin{aligned} r_i(x_t(e)) = \min\{&\min\{t + d(v_s,p_i), d-t+d(v_t,p_i)\} \\ &+\min\{t + d(v_s,q_i), d-t+d(v_t,q_i)\}\} \\ &+ g_i \end{aligned} \tag{26}$$

The following lemma uses (26) to characterize $r_i(\cdot)$.

<u>Lemma 9</u>: For each i, $1 \leq i \leq m$, $r_i(x_t(e))$ is linear in t, $0 \leq t \leq d$, with slope belonging to the set $\{2, 0, -2\}$.

Proof:

<u>Case 1</u>: $x_t(e) \in P_i$

Then either $v_s \in P(x_t(e),p_i)$ or $v_s \in P(x_t(e),q_i)$. Without loss of generality suppose that the former case is true. Then $v_t \in P(x_t(e),q_i)$ and from (26)

$$\begin{aligned} r_i(x_t(e)) &= t+d(v_s,p_i) + d-t+d(v_t,q_i) + g_i \\ &= d(p_i,q_i) + g_i \ . \end{aligned}$$

Thus $r_i(\cdot)$ is linear in t with zero slope.

<u>Case 2</u>: $x_t(e) \notin P_i$

Then since T is a tree either (i) $v_s \in P(x_t(e),p_i(x_t(e)))$ or (ii) $v_t \in P(x_t(e),p_i(x_t(e)))$. If (i) holds true then

$$\begin{aligned} r_i(x_t(e)) &= t + d(v_s,p_i) + t + d(v_s,q_i) + g_i \\ &= 2t + d(v_s,p_i) + d(v_s,q_i) + q_i \end{aligned}$$

and thus $r_i(\cdot)$ is linear with slope 2. In Case (ii) $r_i(x_t(e)) = -2t + d(v_t,q_i) + d(v_t,p_i) + g_i$ which is linear with slope -2. $\square$

Now we define

$$R_i(t) = \max_{1 \leq j \leq i} \{w(j) \cdot r_j(x_t(e))\} . \tag{27}$$

Note that $R_m(t) = r(x_t(e))$ and thus from (25) our objective is to find the minimum of $R_m(t)$ on the interval $(0,d)$.

<u>Corollary</u>: For each i, $1 \leq i \leq m$, $R_i(t)$ is a piecewise linear convex function with slopes belonging to the set $\{\pm 2w(1), \ldots, \pm 2w(i), 0\}$.

Proof: From Lemma 9 each $r_j(\cdot)$ is linear and therefore convex. Thus $w(j)r_j(\cdot)$ is also linear, convex, and has a slope of $\pm 2w(j)$ or 0. Finally it is well known that the maximum of a set of convex functions is convex. □

<u>Remark 6</u>: Since $R_m(t)$ is piecewise linear and convex we could find its minimum by solving a linear programming problem which is equivalent to a two person zero sum bi-matrix game. For the details see [4]. Instead of solving a linear program we will present an iterative method which generates $R_m(t)$ and its minimum values in $O(m \lg m)$ operations.

Next we outline an interative method which can be used to generate R_{i+1} from $R_i(t)$. Since $R_i(t)$ is a piecewise linear convex function let $T_i = \{ 0 = t_o^i, t_1^i, \ldots, t_{n_i}^i = d\}$ be the set of points in the interval $(0,d)$ where the function $R_i(t)$ changes slope. Then for each k, $1 \leq k \leq n_i$ and each $t \in (t_{k-1}, t_k)$ we have,

$$R_i(t) = R_i(t_{k-1}^i) + \frac{R_i(t_k^i) - R(t_{k-1}^i)}{t_k^i - t_{k-1}^i} (t - t_{k-1}^i) . \tag{28}$$

The function $R_i(t)$ is completely described by the list T_i and the values of $R_i(\cdot)$ at the points of T_i. Next, given the list T_i and the values of each member of T_i we outline how $R_{i+1}(\cdot)$ can be generated from $R_i(\cdot)$ or effectively how to generate T_{i+1} and its values from T_i.

From (28) $R_{i+1}(t) = \max\{R_i(t), w(i+1)r_{i+1}(x_t(e))\}$. Now since $w(i+1) \geq \ldots \geq w(1)$ it must be true that $R_i(\cdot)$ and $w(i+1)\cdot r_{i+1}(\cdot)$ either coincide with one another or they intersect in at most two points. Figures 5, 6, and 7 characterize the possibilities which can occur. In (i)-(iv) below we describe how T_{i+1} can be generated from T_i, by taking into consideration the various possibilities which can occur.

Let $\hat{t}(\hat{\hat{t}}) \in [0,d]$ be a point where $w(i+1)\,r_{i+1}(\cdot)$ reaches its minimum (maximum) value. Also let $\bar{t} \in T_i$ be a point where $R_i(\cdot)$ *reaches its minimum value. Let* $T_i(t_i,t_2)$ *be the set of all points in* T_i between t_1 and t_2. We break the different possibilities into four cases:

(i) If $w(i+1)r_{i+1}(x_{\hat{t}}(r)) \geq R_i(\hat{t})$ (Figures 5(a),6(a),7(a)) then

$R_{i+1}(\cdot) = r_{i+1}(\cdot)$ and $T_{i+1} = \{0,d\}$.

(ii) If $w(i+1)r_{i+1}(x_{\hat{\hat{t}}}(e)) \leq R_i(\hat{\hat{t}})$ and $w(i+1)r_{i+1}(x_{\bar{t}}(e)) \leq R_i(\bar{t})$ (Figures 5(c),6(c),7(c)) then

$R_{i+1}(\cdot) = R_i(\cdot)$ and $T_{i+1} = T_i$.

(iii) If $w(i+1)r_{i+1}(x_{\hat{\hat{t}}}(e)) > R_i(\hat{\hat{t}})$ and $r_{i+1}(x_{\hat{t}}(e)) < R_{i+1}(\hat{t})$ (Figures 6(b) and 7(b)) then we find the points $t^i_{k-1} \in T_i$ and $t^i_k \in T_i$ and the unique point $t^* \in (t^i_{k-1},t^i_k)$ such that $r_{i+1}(x_{t^*}(e)) = R_i(t^*)$. Then

$$T_{i+1} = \begin{cases} \{0,t^*\} \cup T_i(t_k^i,d) & \text{if slope} = -2w(i+1) \\ T_i(0,t_{k-1}^i) \cup \{t^*,d\} & \text{if slope} = 2w(i+1) \end{cases} .$$

(iv) $r_{i+1}(x_{\hat{\hat{t}}}(e)) = r_{i+1}(x_{\hat{t}}(e))$; $R_i(\bar{t}) < w(i+1)r_{i+1}(x_{\bar{t}}(e))$; and $w(i+1)r_{i+1}(x_o(e)) < R_i(0)$ or $w(i+1)r_{i+1}(x_d(e)) < R_i(d)$ (Figure 5(b)). Then the either (a) intersect at one point, t^*, $t_{k-1}^i \leq t^* \leq t_k^i$, or (b) they intersect at two points t_1^* and t_2^* where $t_1^* < t_2^*$ and $t_{k-1}^i \leq t_1^* \leq t_k^i$ and $t_{j-1}^i \leq t_2^* \leq t_j^i$. Then

$$T_{i+1} = \begin{cases} T_i(0,t_{k-1}^i) \cup \{t^*,d\} & \text{if (a) and } r_{i+1}(x_o(e)) < R_i(0) \\ \{0,t^*\} \cup T_i(t_k^i,d) & \text{if (a) and } r_{i+1}(x_d(e)) < R_i(d) \\ & \\ T_i(0,t_{k-1}^i) \cup \{t_1^*,t_2^*\} \cup T_i(t_j^i,d) & \text{if (b)} \end{cases} .$$

In cases (i) - (iv) a simple updating procedure has been outlined to generate $R_{i+1}(\cdot)$ from $R_i(\cdot)$. The process is iterative, starting with $R_1(\cdot) = r_1(\cdot)$ and terminating with $R_m(\cdot)$. The interval obtained from the break points in T_m which minimizes $R_m(\cdot)$ provides a set of optimal solutions to P_T. Notice that in cases (i) - (iv) at most two points are added to T_i at any iteration and thus T_m contains at most $2m$ points. Also the lists T_i and their values can be stored in a data structure which is an ordered tree of depth at most $O(\ell g\ m)$. Thus we require at most $O(\ell g\ m)$ operations to retrieve any information which is necessary in (i) - (iv). This makes computational effort involved in the iterative application of (i) - (iv) at most $O(m\ \ell g\ m)$. Combining this with Algorithm 4.1 yields an algorithm for P_T of $O((n+m)\ell gn + m\ell gm)$ of if $m > n$ of $O(m\ \ell g\ m)$.

Example 4: We perform the procedure outlined in Section 4 on the problem in Figure 8 which is identical to the problem in Figure 2 except that positive node weights have been added. We describe the calculations in Algorithm 4.1.

Step 1. $T_0 = T$, $k = 0$

Step 2. Choose node a in Algorithm 4.2 since $N(a) = 4$.

Step 3. $r_1(a) = 2(6 + 2 + 8) = 24$

$r_2(a) = 3(3 + 4 + 7) = 52$

$r_3(a) = 4(3 + 6 + 5) = 56$

so $L(a) = \{3\}$ and $a \notin P_3$. $T_1 = T_b(a)$.

Step 2. Choose node b since $N(b) = 2$ in $T_b(a)$.

Step 3. $r_1(b) = 2(8 + 4 + 8) = 40$

$r_2(b) = 3(5 + 6 + 7) = 54$

$r_3(b) = 4(1 + 4 + 5) = 40$

$L(b) = \{2\}$ and $b \notin P_2$. $T_2 = T_b(a) \cap T_a(b) = (a,b)$.

Step 2. $T_2 = (v_s, v_t) = (a,b)$.

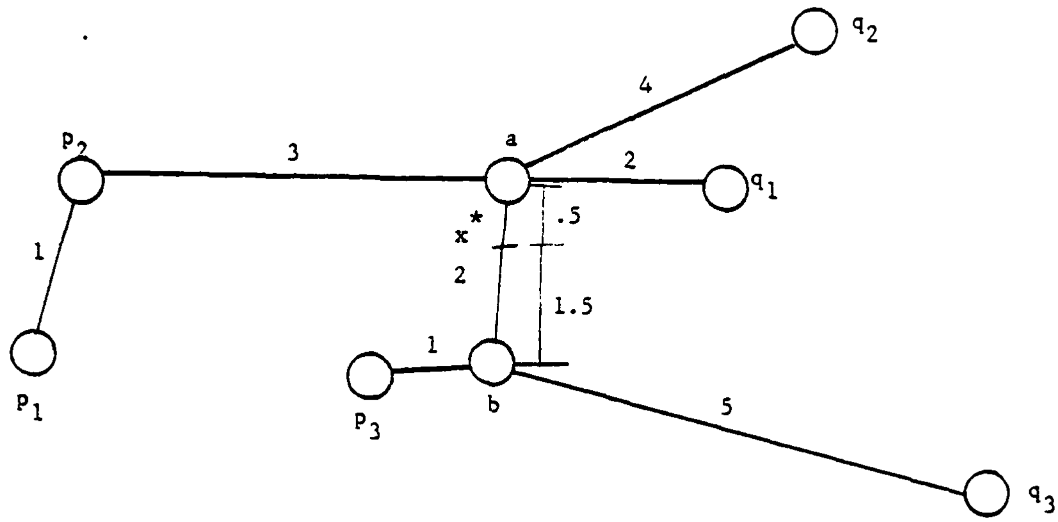
Now we have limited RC to the edge (a,b). Next we apply (i) - (iv) to calculate $R_3(\cdot)$. $R_1(t) = r_1(x_t(e)) = 2t + 32$, from (27) and $T_1 = \{0,2\}$. See Figure 9. Next $r_2(x_t(e)) = 3t + 52$ and this is case (i) and thus $T_2 = \{0,2\}$. Finally $r_3(x_t(e)) = -4t + 56$. This is case (iii), $t^* = \frac{4}{7}$ and $T_3 = \{0,\frac{4}{7},2\}$. $R_3(\cdot)$ is shown in Figure 9. x^* is located on edge (a,b), $\frac{4}{7}$ units from a.

5. Concluding Remarks

In this paper we have shown how the special structure of a tree network can be exploited to provide simple procedures for solving the unweighted and weighted round trip location problems. This includes,

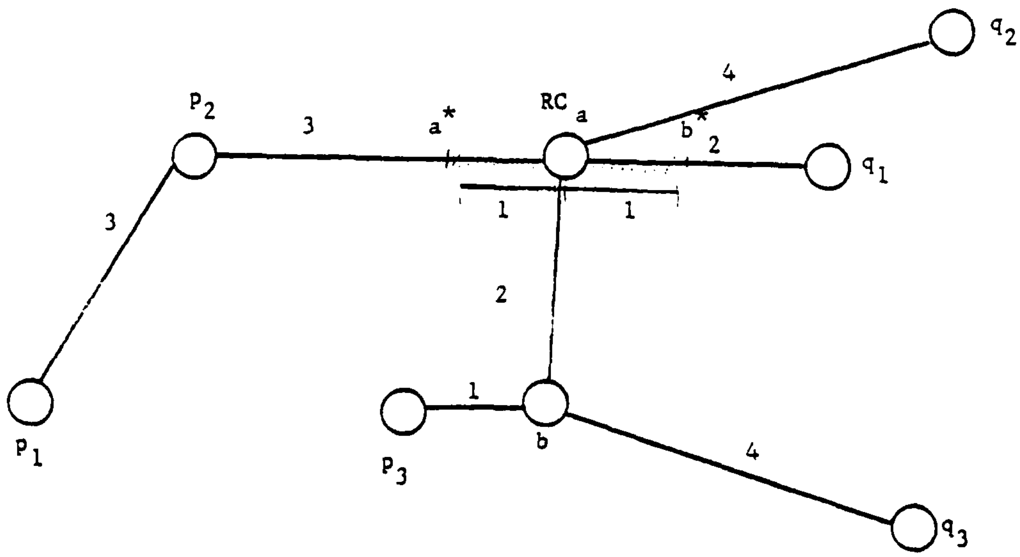
as a special case, the one center problem on a tree. In fact, the method proposed in Section 3 for solving the unweighted version of P_T reduces to the method proposed by Handler for the unweighted one center problem when $p_i = q_i$ and $k_i = 0$ for all i, $1 \leq i \leq m$. Although the results in this paper pertain to the round trip location problem, we feel certain that many of the properties developed here will hold true for several related problems. In particular consider a vehicle routing problem where a vehicle travels from a facility location to a set, S_i, of neighborhoods (or nodes), visiting each neighborhood in S_i exactly once and then returning to the facility. In this paper $S_i = \{p_i, q_i\}$ but in general S_i might contain more than two neighborhoods. Then one might consider locating a facility so as to minimize the total cost of visiting all of the neighborhood sets or to minimize the maximum cost of visiting any neighborhood set. To our knowledge very little work has been done on combined vehicle routing-location problems.

Finally we would like to point out what we consider to be the crucial property of the unweighted version of P_T which justifies the $O(m)$ algorithm described in Section 4.3. Lemma 8 of Section 2 says that for each $x \in T$ the set of optimal solutions to P_T intersects the path from x to each of its bottleneck node pairs. This property no longer holds true for the weighted version of P_T. Consider the simple three node example in Figure 10. Here $p_i = q_i$ and $k_i = 0$ for all i. Consider node 1, $r_2(1) = 2$ and $r_3(1) = 4$; thus $r(1) = 100 \cdot 2 = 200$ and $L(1) = \{2\}$. However the unique optimal solution, x^*, is located between nodes 2 and 3, .2 units from node 2.



$w(i) = 1 \ \forall i \quad k_i = d(p_i, q_i)$

Figure 1



$w(i) = 1 \ \forall i \quad k_i = d(p_i, q_i)$

Figure 2

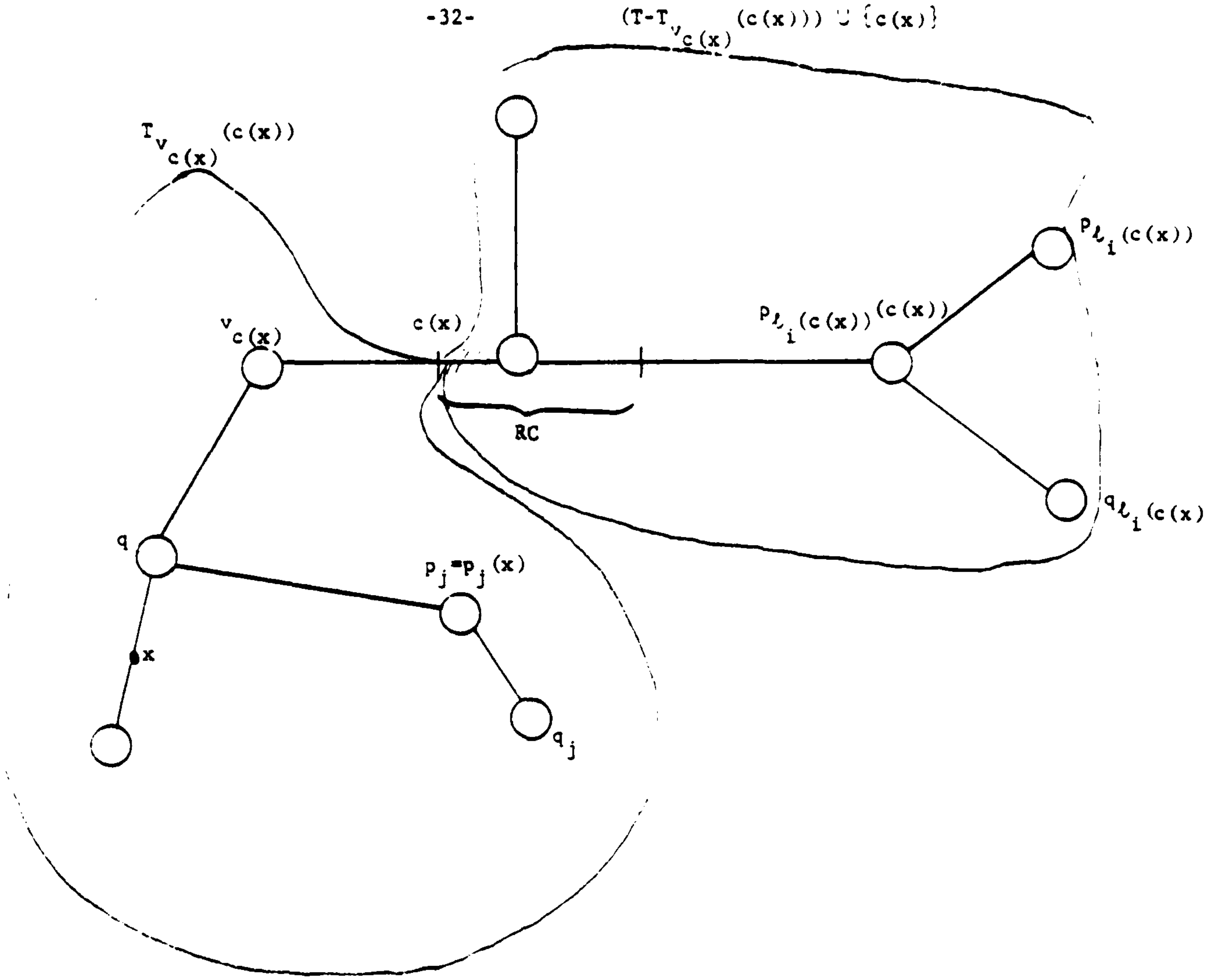


Figure 3

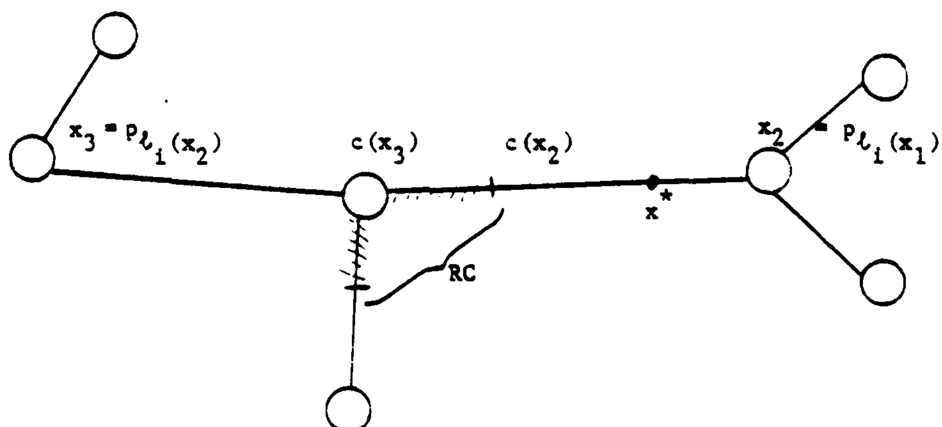


Figure 4

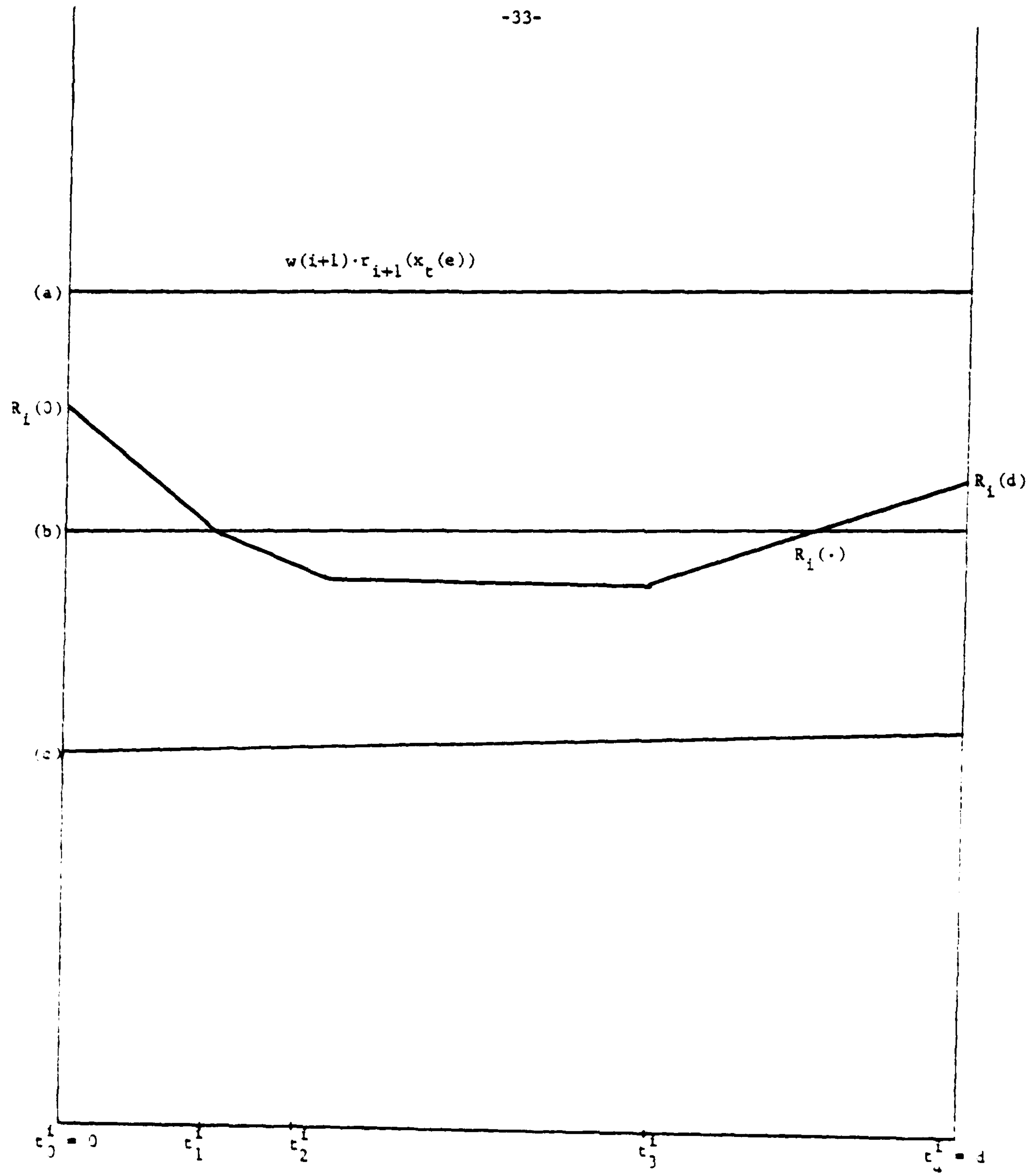


Figure 5

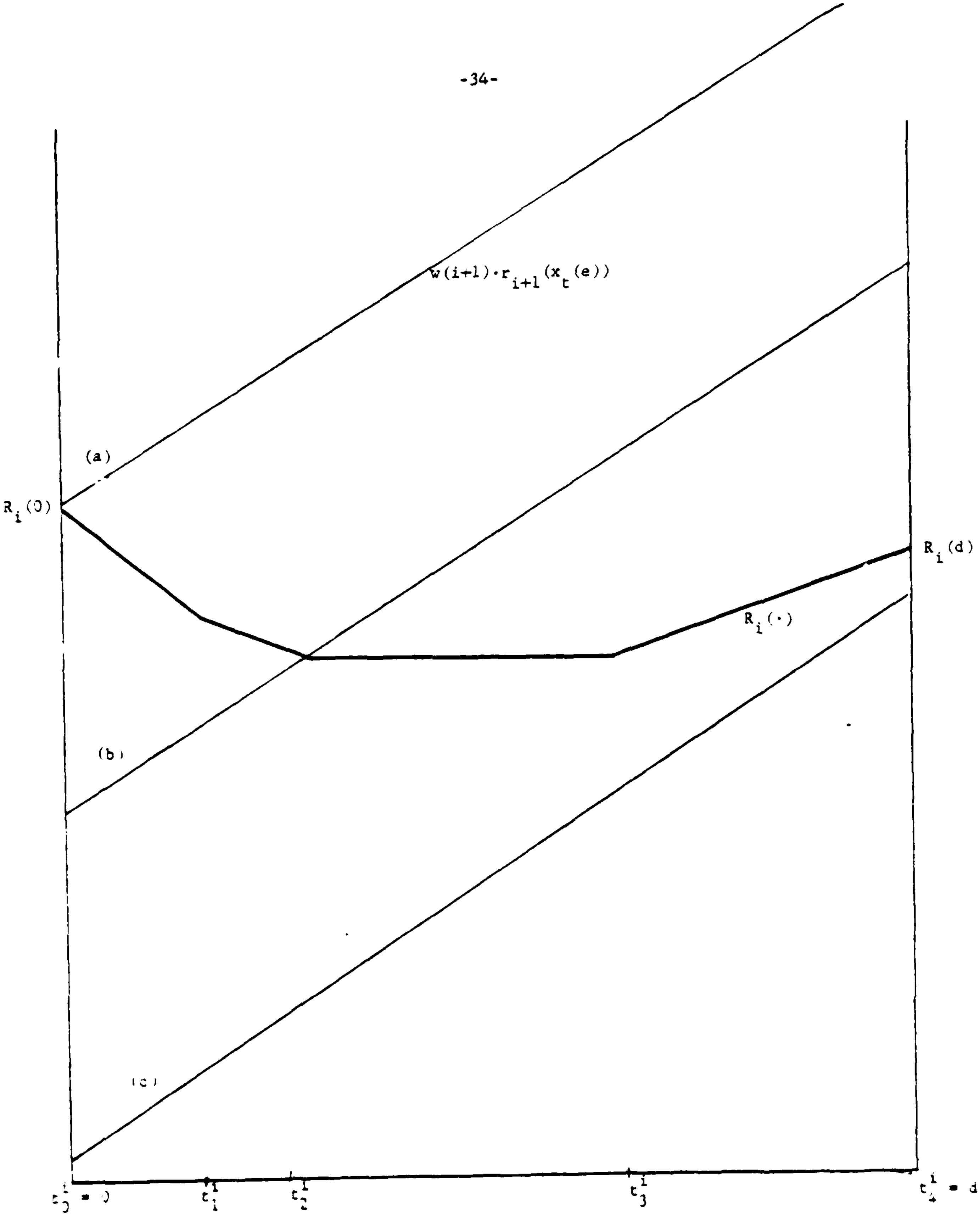


Figure 6

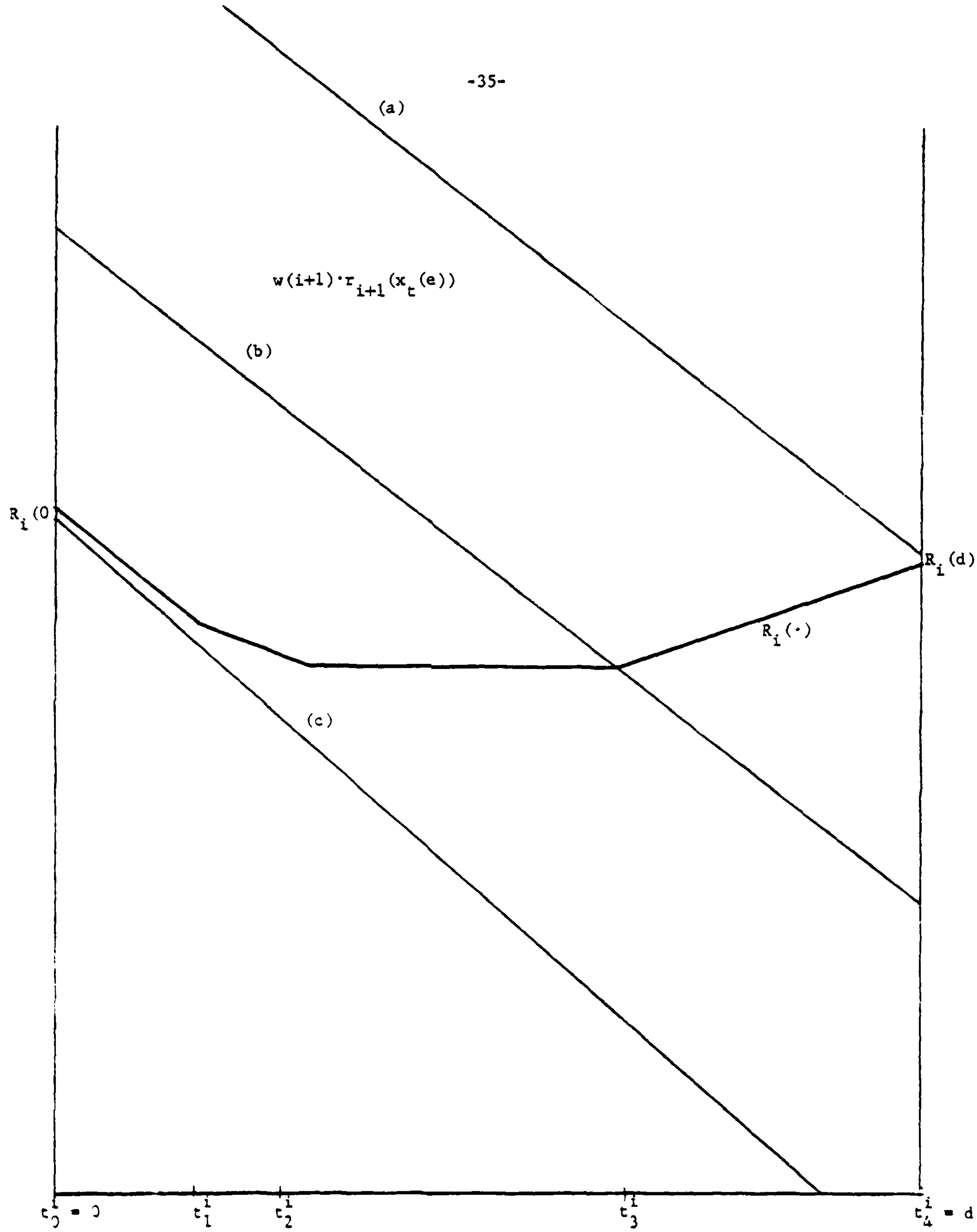
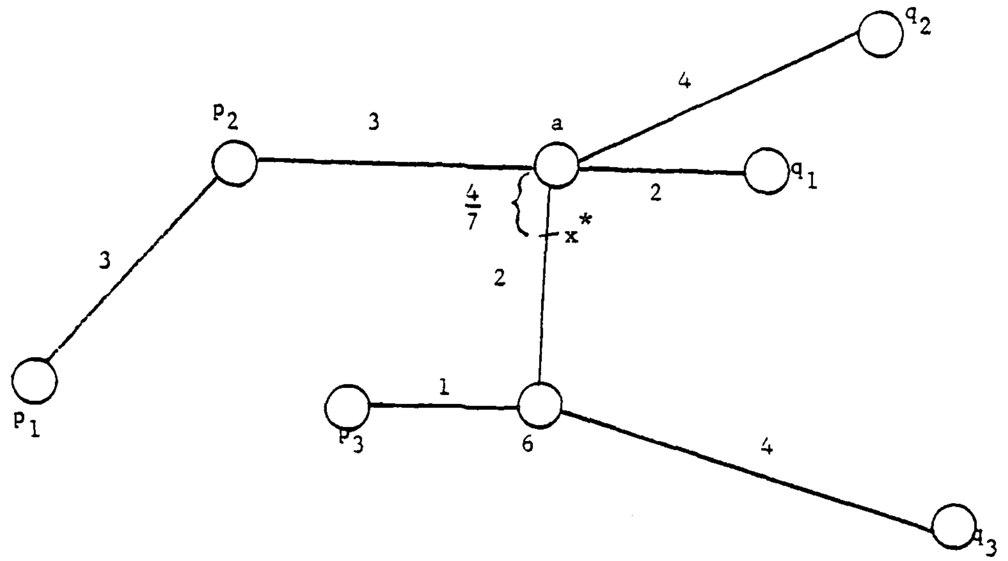


Figure 7



$w(1) = 2$ $w(2) = 3$ $w(3) = 4$

$k_i = d(p_i, q_i)$

Figure 8

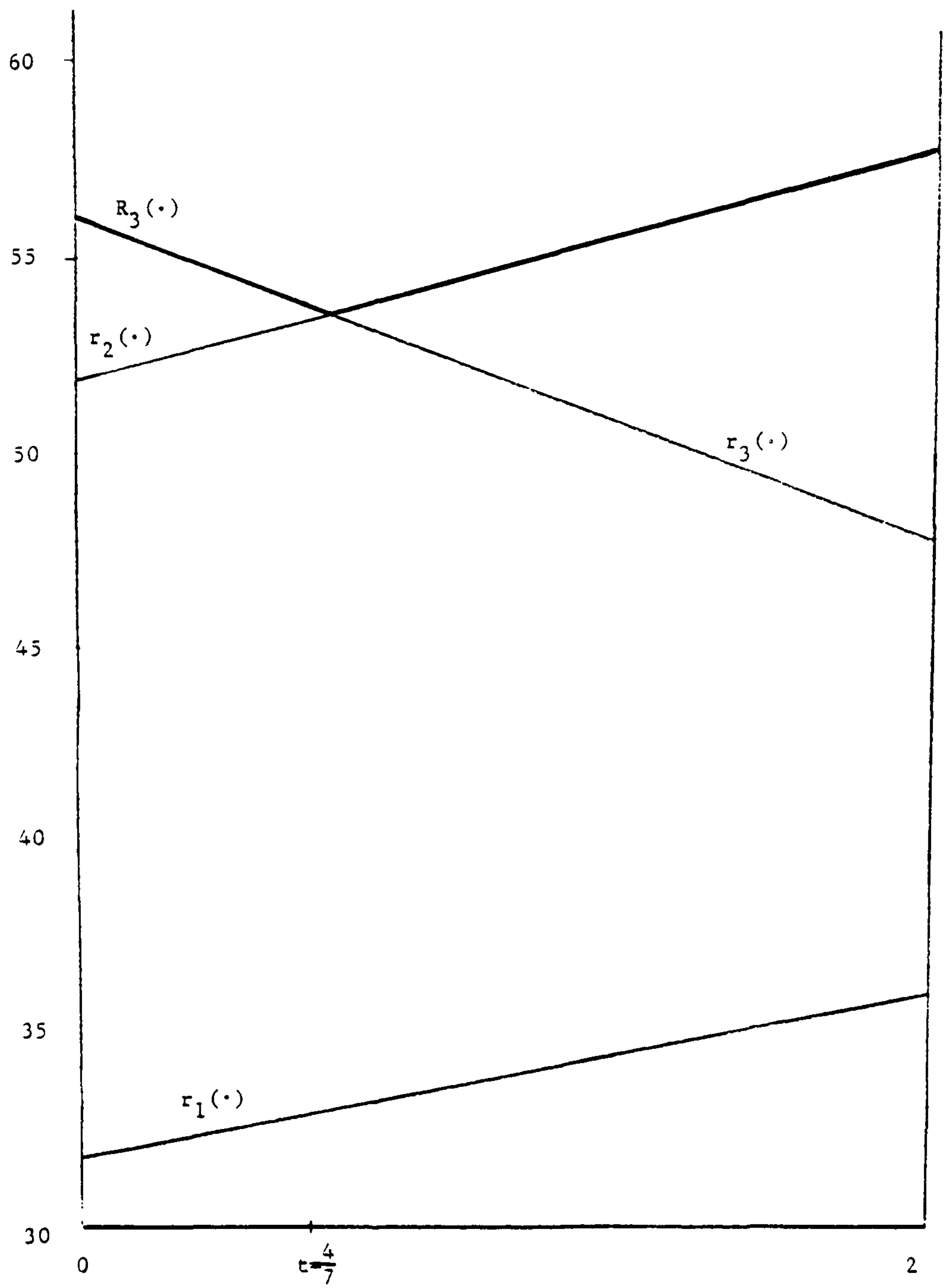


Figure 9

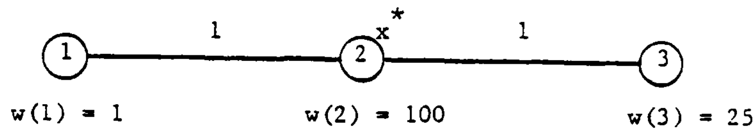


Figure 10

References

1. Chan, A. and R. L. Francis, "A Round Trip Location Problem on a Tree Graph," Transportation Science, 10, pp. 35-51, (1976).

2. Chan, A. and D. W. Hearn, "A Rectilinear Distance Minimax Round Trip Location Problem," Transportation Science, 11, pp. 107-123 (1977).

3. Dearing, P. M., R. L. Francis and T. J. Lowe, "Convex Location Problems on Tree Networks," Operations Research, 24, 628-642, (1976).

4. Gaver, D. and G. L. Thompson, Programming and Probability Models in Operations Research, Brooks/Cole (1973).

5. Goldman, A. J., "Minimax Location of a Facility in a Network," Transportation Science, 6, pp. 407-418, (1972).

6. Hakimi, S. L. and O. Kariv, "An Algorithmic Approach to Network Location Problems, Part I: The p-Centers," Working Paper, Northwestern University.

7. Handler, G. Y., "Minimax Location of a Facility in an Undirected Tree Graph," Transportation Science, 7, pp. 287-293, (1973).